International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)
ISSN(P): 2249-6890; ISSN(E): 2249-8001
Vol. 8, Issue 1 Feb 2018, 681-690

A RESTART SCHEME BASED META HEURISTIC FOR FLEXIBLE JOB SHOP SCHEDULING

RAJAN & VINEET KUMAR

Department of Mechanical Engineering, University Institute of Engineering & Technology, Maharishi Dayanand University, Rohtak, Haryana, India

ABSTRACT

Flexible job shop scheduling problem (F.J.S.S.P) is very important in the fields of production management and it appears under the category of N.P hard combinatorial optimization problem. It is an expansion of traditional job shop scheduling problem (J.S.S.P), though, in many organizations, schedules are mandatory with the existence of various diverse sudden interferences. That's why, it's very complicated to have best possible results within reasonable time. Genetic algorithm system (G.A.S) can reduce combinatorial complexity by task breakdown & real time allotment methods. Genetic algorithm system (G.A.S) and human immune system (H.I.S) are analogous in genetic structure and negotiation strategies. Moreover, structure and negotiation strategies of G.A.S are inspired by H.I.S and are much reliable with the negotiation strategies of H.I.S.. The purpose of this paper is to optimize F.J.S.S.P using G.A.S. In total, a case study has been considered to access the performance of F.J.S.S.P with an objective to reduce make span (C_{max}). A restart scheme is entrenched into regular G.A.S for avoiding premature convergence and hence improvement in the fitness value. Randomly selected process plan results into improved shop performance.

KEYWORDS: Genetic Algorithm, Production Scheduling & Flexible Job Shop Scheduling Problem (F.J.S.S.P)

Received: Dec 02, 2017; **Accepted:** Dec 22, 2017; **Published:** Jan 19, 2018; **Paper Id.:** IJMPERDFEB201874

Original Article

1. INTRODUCTION

In order to produce several goods, machines (resources) inside production plant, execute some actions on shop floor. The time required for execution of these actions is known as process time. This dealing leads to erect limitations like, resource preparation time, release date and job (task) induction time [6]. Due to these limitations and in order to satisfy the constraints, requirement of scheduling arises for computing the best process time of operations on machines. This forecasting generally comprises a category of problem, known as job shop scheduling problem (J.S.S.P) [5]. In a job shop, orders of goods to be made are in less quantity. Flow of work is multidirectional. Every resource is categorized by input/output work flow and 'n' number of tasks is performed on 'm' number of resources. Every task 'j' comprises 'n_j' operations (processes). J.S.S.P presumes single eligible resource for every process and viable single process plan (process sequence) for every task. Every process '$O_{j,i}$' of task 'j' must be operated by one resource amongst the set of entitled machines '$M_{j,i}$'. Flexibility in job shop can be attained by installing multiple eligible resources and multiple process plans on shop floor; hence, job shop gets the extension name as flexible job shop [28]. A shop is referred as totally flexible, if every resource can perform all processes; if not, it is partially flexible. Processing time '$p_{j,i,k}$' of process '$O_{j,i}$' on a resource 'k' ϵ '$M_{j,i}$' is constant. For uninterrupted processing, every resource can process a particular task at a particular time. Processing of '$O_{j,(i+1)}$' cannot begin until the processing of '$O_{j,i}$' gets finished i.e., pre-emption is illegal. Proposed work accounts

scheduling in static as well as in flexible environment, where tasks are ready at zero time (t = 0).

F.J.S.S, which is N.P hard in nature, is more complex than classical J.S.S [22]. Because of addition of multiple process plans in F.J.S.S, complexity increases. This forced the researchers to focus on various meta heuristic techniques, like: genetic algorithm (G.A) [7], tabu search (T.S) [9], simulated annealing (S.A) [27], particle swarm optimization (P.S.O) [14], fuzzy logic [17] etc. Present work proposes a modified version of G.A for F.J.S.S.P, because, it is very efficient in having best possible results. The goal is to evaluate fitness value using mathematical functions. Make span 'C_{max}' is considered as performance measure. Make span stands for time required to finish all tasks [25].

$C_{max} = \max_{1 \leq i \leq n} x\ C_i$

Where:

C_i = finishing time of task 'j_i'.

Proposed modified G.A consists of (i) an effective selection method called "tournament selection", (ii) a latest crossover operator, which uses hierarchical cluster model to collect population in each generation and (iii) a fresh mutation operator, which helps to maintain population diversity and overcomes premature convergence [2].

This paper is categorized as follows. Section 2 presents literature review on F.J.S.S. Section 3 presents sample problem as definition of F.J.S.S.P. Detailed flow chart for G.A.S as methodology adopted, presents, in section 4. Section 5 gives description of results and section 6 finally draws the conclusion and future work.

2. REVIEW OF LITERATURE

In literature, two vital modes have been considered to resolve F.J.S.S.P by using G.A. One is how to encode a solution of F.J.S.S.P for getting a feasible solution [1]. Second is how to represent and deduct schedule in order to reduce make span by employing a heuristic technique. Reference [20] intended a 2-row offspring arrangement based on working system and resource allocation for reducing make span. F.J.S.S.P is N.P hard in nature, it is impossible to crack it efficiently with single technique, hence; vigorous research has focussed on different schemes of integration [4]. A framework is proposed [19], how to find out the arrangement of task release order and sequence of dispatching the tasks on every resource. Reference [32] used multi step crossover fusion (M.S.X.F) as unified operator and recombination operator in G.A to solve bulky problems. A heuristic, G.A, is developed based on mixing technique for solving F.J.S.S.P with parallel resources and pre-emption constraints for reducing make span [10]. Proposed method obtained improvement for longer running time. For attaining best possible task completion time, batch size and process series are accounted as variables [30]. It concludes that task completion time can be minimised if batch size is taken into account. Reference [11] introduces G.A to describe priority of processes & delay time. Author integrates parameterise active schedule and local search heuristic, in order to improve ultimate result. Customized G.A was offered [29] to solve J.S.S.P using process based depiction and represent the results in active schedule through search processes. Author replaced classical mutation of G.A with metropolis sample process of S.A, in hybrid manner. In order to evaluate tardiness, release dates & due dates are accounted [21]. Here, limitation of G.A i.e., searches for increasing problem size, can be moderate by tuneable schedule builder and multistage decomposition approach. In order to calculate minimum make span, a heuristics job order integrates with G.A for small to medium sized F.J.S.S.P [12]. For generating initial individuals, reference [3] used hybridization of G.A with other heuristics, like, shortest processing time (S.P.T), minimum slack (M.S) and longest processing time

(L.P.T). Reference [33] advised a hybrid algorithm for reducing weighted tardiness in which G.A determined first process of every resource while heuristics determined assignment of left over processes. For F.J.S.S, in a distributed production environment, reference [16] combines G.A and gantt chart (G.C) in order to find out process plans of small and medium size problems. G.A shows parallelism, encloses historical data of previous results and is appropriate for execution on huge parallel designs [29].

Literature discloses various G.A techniques that have been employed to crack F.J.S.S.P. All these techniques vary from each other in terms of representation & operators handled constraints used and objectives attained. Inspite of all dissimilarities, G.A has been certified as well organized meta heuristic for cracking F.J.S.S.Ps. The literature reveals that restart scheme based G. A has not been accounted deeply till yet, for F.J.S.S.Ps. Restart scheme avoids premature convergence and finally improves fitness value [26]. Thus, restart scheme based G.A can be taken as an emphatic approach in order to deal with F.J.S.S.Ps.

3. SAMPLE PROBLEM

In F.J.S.S.P, the main aim is to execute 'n' number of tasks on 'm' number of resources. Set of resources is noted as 'k', {M1, M2, ..., Mm}. Each task 'j' consists of a sequence of 'n_j' processes and every process '$O_{j,i}$', i.e., 'i^{th}' process of 'j^{th}' task requires one resource out of a set of given resources called '$M_{j,i} \subseteq k$'. The problem is to select a sequence of processes together with the assignment of start/end times and resources for each process. In this paper, a criterion 'F1' is to be reduced.

F1: To reduce make span 'C_{max}' i.e., to reduce the maximum completion time of resources.

A sample problem is given on F.J.S.S. in Table 1. Every row cites a process; each column deals a resource and individuals in every cell denote processing time. As demonstrated, 2nd process of 3rd task is allowed only to be processed on M2, M4 and M5. Symbol "-" justifies that resource cannot perform consequent process. To explain F.J.S.S.P, following case study has been considered from reference [31]. Prime motive is to execute four tasks on six resources, according to processing time.

Table 1: Sample Problem of Flexible Job Shop Scheduling

Job	Operations	Machines: M1	M2	M3	M4	M5	M6	Operation No.
	O1,1	2	3	4	-	-	-	1st Operation
J1	O1,2	-	3	-	2	4	-	2nd Operation
	O1,3	1	4	5	-	-	-	3rd Operation
	O2,1	3	-	5	-	2	-	
J2	O2,2	4	2	-	-	6	-	
	O2,3	-	-	4	-	7	10	
	O3,1	5	6	14	-	-	-	
J3	O3,2	-	4	-	2	5	-	Time in Minutes
	O3,3	-	-	13	-	8	12	
	O4,1	9	-	7	9	-	-	
J4	O4,2	-	6	-	4	-	5	Total no. of Operations = 12
	O4,3	1	-	3	-	-	3	
		where:	Sample Problem of size : n x m (4 x 6) n = no. of jobs m = no. of machines					

Following hypotheses for F.J.S.S.P, accounted in proposed work has been taken from [13].

- All tasks are independent from each other and can be release at time t = 0;
- All resources are independent from each other and available at time t = 0;
- Sequence of processes is predetermined for every task and can't be modify;
- At a given time, each resource can execute single process: it becomes available to other processes if the process which is processing gets completed;
- Setting up times of resources and move times between processes are negligible;
- Each process '$O_{j,i}$' must be processed without interruption on any resources.
- A task cannot depart the shop floor before all its processes get completed.
- Order cancellation, rework and resource breakdown are not allowed.
- S.P.T rule is used in order to process all tasks.

3.1. Variables & Prime Objective

All 'i', 'j' and 'k' are named as below:

'j' = 1,...,n; 'i' = 1,....,n_j ; 'k' = 1,....,m.

Where:

'n' = sum of tasks,

'n_j' = sum of operations of task 'j' and

'm' = sum of resources.

Variables:

'$p_{j,i.k}$' = processing time (priori) of operation 'i' on task 'j' at resource 'k';

'F_k' = finishing time of resource 'k';

'$O_{j,i}$' = processing time (priori) list of process 'i' on task 'j';

'$C_{j,i}$' = antigen (tasks/job) concentration value i.e., shortest processing time of uncompleted process 'i' on task 'j';

$$C_{j,i} = \sum_{i}^{nj} \min.(0j, i)$$

Where: '$O_{j,i}$' = {$p_{j,i,1}$,.....,$p_{j,i,m}$}

Main objective: To minimize makespan (F1)
min. F1 = min. (max. (F_k))

4. METHODOLOGY ADOPTED

Proposed work exploits a 'restart scheme' based G.A.S as follows. Each step is detailed in subheadings.

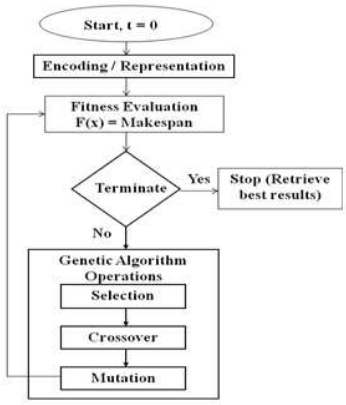

Figure 1: Main Stages of a Basic Genetic Algorithm System (G.A.S)

4.1. Encoding / Representation of Chromosome

It is first step of G.A and is sequence oriented by nature. Part type process plan number (alphabet) is used to generate a thread/gene of a chromosome which is in fixed ordered pair (tuple). Every gene is transformation of triplet (j, i, k). Where: j, i and k stands for task, process and resource, respectively. A tuple hints that process '$O_{j,i}$' will be treated on resource 'k'. Every tuple site decides the priority of consequent process. Left cornered tuple has prime consideration. Suppose, there are three part types multiple process plans (M.P.P) namely, 'A', 'B' and 'C', respectively. Information can be encoded as: {4, 1, 2}.

Where:

- '4' represents processing of 'A' on pursuing fourth process plan,
- '1' represents processing of 'B' on pursuing first process plan and
- '2' represents processing of 'C' on pursuing second process plan.

Proposed work accounts random generation of initial population (N_{pop}).

4.2. Fitness Assignment and Selection

In G.A two operators are available for selection, namely, (i) mating operator and (ii) environmental operator. Former selection operator justifies that which individual chromosome will do crossover in order to breed offspring and latter selection operator identifies that which individual chromosome will stay alive in next generation. Chromosome with best fitness value undergoes the linear ranking (L.R) selection method with stochastic universal sampling (S.U.S). Amongst various assignment methods this paper refers [8] as the best fitness assignment method. Present work proposes 'tournament' type mating selection method, in which two individuals are selected, randomly. Once anticipated values got allotted, S.U.S is implemented in order to choose parents. Finally, a mating pool is formed which consists selected individuals only [22].

4.3. Crossover Operator

In proposed work, 'two point' crossover technique is accounted which has to be implemented on chosen individuals from mating pool. In order to make a pair, two off springs are taken into account, randomly, from mating pool with crossover probability 'p_c' = 0.8. Location is selected, randomly, for two times, from starting to end position.

Crossover operator should be able to swap routing & sequencing decisions because parental chromosome encloses both the necessary information. Assignment crossover operator (As. X) swaps the resources of randomly selected processes. While, precedence preserving order based crossover operator (P.P.O.X) [18] chooses single task, randomly and maintains the positions of its processes, unaltered, in every parents. All leftover processes on every parent chromosome are relocated in the order they appear in another parent chromosome. Amongst 'As. X' & 'P.P.O.X', one chromosome is chosen, randomly, for breeding the offspring during crossover.

4.4. Mutation Operator

Mutation operator is implemented on new generated chromosome after crossover process. Proposed work uses 'mutual swap' with probability 'p_m' = 0.2. Mutation sites are selected, randomly, for two times from starting to last gene. Process plans at these sites are swapped, retaining another genes unaltered. In order to regulate resource assignment, two heuristic mutation operators are utilized. These operators recognize the resource 'm_{max}' with maximum workload (make span) and then randomly select a process '$O_{j,i}$' on 'm_{max}'. Process '$O_{j,i}$' is assigned to the resource 'm_{min}' having minimum workload (make span). If 'm_{min}' is not eligible for process '$O_{j,i}$', then '$O_{j,i}$' is allotted to another resource, randomly, out of adequate set of given resources '$M_{j,i} \subseteq k$'.

There is possibility that some illicit (illegal) offspring may engender after mutation operator. One part type might go beyond the limit of specified process plan out of those individuals which undergo mutation. Therefore, repairing of offspring is needed in order to eliminate developed illegality. Repairing also verifies, if any particular site of string goes beyond the specific limit then that particular site shall be substituted by those process plan which is lacking from the given limit. For reproduction, elitism is nested with L.R selection method. Elitism rate = 0.9 is accounted for shifting best individuals from previous to next generated population. Present work suggests the termination criteria as maximum number of generations 'n x m' [23, 24].

Where:

'n' = number of tasks and

'm' = number of resources.

A restart scheme based G.A is applied [26] in order to avoid premature convergence. Hence, best fitness value is stored at every generation. A counter is set for maximum value of pre-defined number of generations (best count). If best fitness value does not alter for more than a pre defined number of generations then proposed algorithm will restart to regenerate the population.

A restart scheme will work as follows:

- Arrange all fitness values in descending order;
- Omit first 25% individuals from arranged record;

Leftover 75% individuals from arranged record will be regenerated as follows:

- Engender 50% of new chromosomes amongst first 25% individuals by mutual exchange;
- Breed 50% of new chromosomes by random selection.

Every latest engendered chromosome will take over 75% poor individual from whole inhabitants if they will return best results of fitness function as compare to poor individual of preceding inhabitants. Reiteration is inadmissible amongst latest engendered chromosomes.

5. DESCRIPTION OF RESULTS

F.J.S.S.P containing four jobs on six machines is accounted for implementing G.A as adopted methodology. Table 2 presents the results of sample problem used for execution. Transportation time in minutes is presumed between all the machines. Make span is treated as the performance criteria. According to least production time criteria, sample problem deals with randomly selected single process plan (S.P.P) of part type.

Proposed work believes in adequacy of G.A.S with restart scheme. Figure 2 confirms convergence curve between fitness values versus number of generations. Curve shows average fitness value appears to be 0.071428 that avoids premature convergence and hence improves fitness value.

Table 2: Computational Results from Proposed Algorithm

No. of Generations (n x m)	Fitness Value (1/fx)	No. of Generations (n x m)	Fitness Value (1/fx)
1	0.052631	13	0.0625
2	0.052631	14	0.066666
3	0.052631	15	0.066666
4	0.052631	16	0.066666
5	0.052631	17	0.066666
6	0.052631	18	0.066666
7	0.055555	19	0.066666
8	0.058823	20	0.071428
9	0.0625	21	0.071428
10	0.047619	22	0.071428
11	0.0625	23	0.071428
12	0.05	24	0.076923

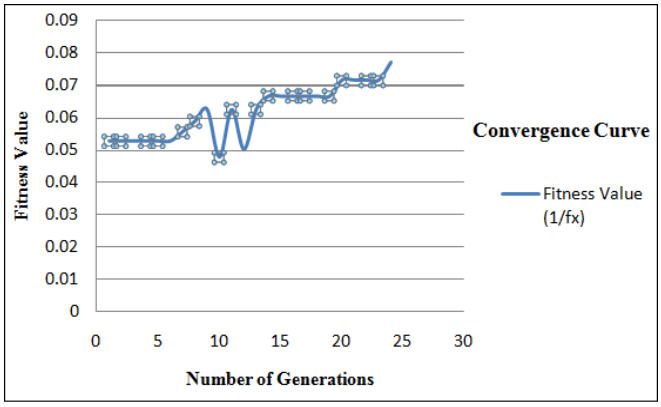

Figure 2: Convergence Curve of G.A.S with Restart Scheme

6. CONCLUSIONS AND FUTURE WORK

Thanks to earlier researchers, proposed work covers the possibility that a restart scheme based G.A has been implemented efficiently to solve F.J.S.S.P. As performance criteria, make span is accounted. Results signify that randomly selected part type process plan comprises improved quality. On entrenching restart scheme in G.A.S, value of fitness

function improves. For attaining further improvements in current solution, development of a new algorithm is required in order to calculate the performance measure through hybridization of two meta heuristic techniques, broadly.

REFERENCES

1. *Amirthagadeswaran, K.S. and Arunachalam, V.P. (2006), "Improved solutions for job shop scheduling problems through genetic algorithm with a different method of schedule deduction", The International Journal of Advanced Manufacturing Technology, Vol. 28 No. 5, pp. 532-40.*
2. *Baker, J.E. (1985), "Adaptive selection methods for genetic algorithms", Proceeding on the First International Conference on Genetic Algorithms and their Applications, Lawrence-Erlbaum, Mahwah, NJ, pp. 101-11.*
3. *Buzatu, C. and Bancila, D. (2008), "A hybrid algorithm for job shop scheduling", Proceedings of 6th International DAAAM Baltic Conference, INDUSTRIAL ENGINEERING, Tallinn, Estonia, 24-26 April.*
4. *Cheng, R., Gen, M. and Tsujimura, Y. (1999), "A tutorial survey of job shop scheduling problems using genetic algorithms. Part II: hybrid genetic search strategies", Computer and Industrial Engineering, Vol. 36, pp. 343-64.*
5. *N. Jananeeswari et al., Multi Objective for a Partial Flexible Open Shop Scheduling Problems using Hybrid Evolutionary Algorithm, International Journal of Mechanical and Production Engineering Research and Development (IJMPERD), Volume 7, Issue 1, January - February 2017, pp. 1-12*
6. *Conway, R.W., Maxwell, W.L. and Miller, L.W. (1967), Theory of Scheduling, Addison-Wesley, Reading, MA.*
7. *D.C. Mattfeld, Evolutionary Search and the Job Shop, Physica-Verlag, Berlin, 1996.*
8. *D. E. Goldberg, Genetic Algorithms in Search Optimisation and Machine Lerning, Addison-Wesley, Reading, Massachusetts, 1989.*
9. *Deb, K., Pratap, A., Agarwal, S., Meyarivan: A Fast and Elitist Multi objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182 –197 (2002).*
10. *F. Glover, Tabu Search-Part I, Operations Research Society of America, J. Comput., 1989.*
11. *Ghedjati, F. (1999), "Genetic algorithms for the job-shop scheduling problem with unrelated parallel constraints: heuristic mixing method machines and precedence", Computers & Industrial Engineering, Vol. 37 No. 1, pp. 39-42.*
12. *Goncalves, J.F., Magalhaes, J.J., de Rua, D., Frias, R., Jose, J., Mendes, M. and Resende, M.G.C.(2002), "A hybrid genetic algorithm for the job shop scheduling problem", AT&T Labs Research Technical Report TD-5EAL6J, September.*
13. *Hasan, S.M.K., Sarker, R. and Cornforth, D. (2007), "Hybrid genetic algorithm for solving job-shop scheduling problem", 6th IEEE/ACIS International Conference on Computer and Information Science, ICIS, pp. 519-24.*
14. *I. Kacem, S. Hammadi, and P. Borne, Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transaction on Systems, Man and Cybernetics, (2002) 32(1), 1-13.*
15. *J. Kennedy and R. Eberhard, Particle Swarm Optimization, Phys. Rev., B. (1995) 13:5344-5348.*
16. *Jain, A.S. and Meeran, A. (1999), "A state-of-the-art review of job-shop scheduling techniques", European Journal of Operations Research, Vol. 113, pp. 390-434.*
17. *Jia, H.Z., Fuh, J.Y.H., Nee, A.Y.C. and Zhang, Y.F. (2007), "Integration of genetic algorithm and Gantt chart for job shop scheduling in distributed manufacturing systems", Computers & Industrial Engineering, Vol. 53 No. 2, pp. 313-20.*
18. *L. Zadeh, Fuzzy sets, Information and Control, 8(3) (1965) 338-353.)*

19. *Lee, K.M., Yamakawa, T., Lee, K.M.: A Genetic Algorithm for General Machine Scheduling Problems. In: Proceedings of International Conference on Knowledge-based Intelligent Electronic Systems, pp. 60 –66 (1998).*
20. *Lee, C.Y., Piramuthu, S. and Tsai, Y.K. (1997), "Job shop scheduling with a genetic algorithm and machine learning", International Journal of Production Research, Vol. 35, pp. 1171-91.*
21. *Li, Y. and Chen, Y. (2010), "A genetic algorithm for job-shop scheduling", Journal of Software, Vol. 5 No. 3, pp. 269-74.*
22. *Mattfeld, D.C. and Bierwirth, C. (2004), "An efficient genetic algorithm for job shop scheduling with tardiness objectives", European Journal of Operational Research, Vol. 155 No. 3, pp. 616-30.*
23. *Mitchell, M. (2002), An Introduction to Genetic Algorithms, Prentice-Hall, New Delhi.*
24. *Ponnambalam, S.G., Ramkumar, V. and Jawahar, N. (2001), "A multi objective genetic algorithm for job shop scheduling", Production Planning & Control, Vol. 12 No. 8, pp. 764-74.*
25. *Rajan and Vineet Kumar, "Hybridization of genetic algorithm & variable neighborhood search (VNGA) approach to crack the flexible job shop scheduling problem: A framework, "Journal of Industrial Pollution Control, vol. 33, no. 2, p.p 90-99, 2017.*
26. *Rajan and Vineet Kumar, "Flow Shop & Job Shop Scheduling: Mathematical Models," International Journal of R&D in Engineering, Science and Management, vol. 5, no. 7, p.p 1-7, 2017.*
27. *Ruiz, R. and Maroto, C. (2006), "A genetic algorithm for hybrid flow shops with sequence dependent setup times and machine eligibility", European Journal of Operational Research, Vol. 169 No. 3, pp. 781-800.*
28. *S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, Sci., vol. 220 (1983) 671-680.*
29. *Vinod, V. and Sridharan, R. (2008), "Scheduling a dynamic job shop production system with sequence-dependent setups: an experimental study", Robotics & Computer-Integrated Manufacturing, Vol. 24, pp. 435-49.*
30. *Wang, L. and Zheng, D.Z. (2002), "A modified genetic algorithm for job shop scheduling", International Journal of Advance Manufacturing Technology, Vol. 20, pp. 72-6.*
31. *Xie, H. (2001), "A genetic algorithm approach to job shop scheduling problems with a batch allocation issue", 17th International Workshop on Artificial Intelligence, Seattle, WA, 1 August, pp. 126-31.*
32. *Y. R. Zou, "A flexible job shop pre/re-scheduling system based on agent". The Economics and Management School, Beijing: Beijing University of Technology, 2011.*
33. *Yamada, T. and Nakano, R. (1997), "Genetic algorithms for job-shop scheduling problems", Proceedings of Modern Heuristic for Decision Support, London, 18-19 March, p. 67.*
34. *Zhou, H., Cheung, W. and Lawrence, C.L. (2009), "Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm", European Journal of Operational Research, Vol. 194 No. 3, pp. 637-49.*